Net-Centric Implementation Framework

Part 1: Overview

Part 2: ASD(NII) Checklist Guidance

Part 3: Migration Guidance

Part 4: Node Guidance

Part 5: Developer Guidance

Part 6: Acquisition Guidance

V 1.3

16 June 2006

NESI (Net-Centric Enterprise Solutions for Interoperability) is a collaborative activity of the USN PEO for C4I and Space, the USAF Electronic Systems Center, and the Defense Information Systems Agency.

Table of Contents

1 NESI Implementation

NESI Part 2: ASD(NII) Checklist Guidance is the second of six parts of the NESI implementation document set. Part 2 is intended for managers of new programs or programs that are undergoing a transformation or major upgrade. Use especially in the pre-systems acquisition and systems acquisition phases. Reference (n) uses net-centric design precepts called **tenets** to guide the move into the net-centric environment. NESI provides specific technical direction for satisfying reference (n). Note that some tenets address doctrinal or procedural requirements; this guidance does not address those areas.

This section contains NESI background information. For a more complete overview of NESI, see the first part of this document set, *NESI Part 1: Overview*.

1.1 References

(a) DoD Directive 5000.1, *The Defense Acquisition System*, 24 November 2003.

(b) DoD Instruction 5000.2, *Operation of the Defense Acquisition System*, 12 May 2003.

(c) DoD Directive 8100.1, *Global Information Grid (GIG) Overarching Policy*, 21 November 2003.

(d) DoD Directive 4630.5, *Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS)*, 05 May 2004.

(e) DoD Instruction 4630.8, *Procedures for Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS)*, 30 June 2004.

(f) DoD Directive 5101.7, *DoD Executive Agent for Information Technology Standards*, 21 May 2004.

(g) *DoD Global Information Grid (GIG) Architecture, Version 2.0*, August 2003.

(h) *DoD Architecture Framework (DoDAF), Version 1.0*, 9 February 2004.

(i) *DoD Net-Centric Data Strategy*, DoD Chief Information Officer, 9 May 2003.

(j) CJCSI 3170.01E, *Joint Capabilities Integration and Development System*, 11 May 2005.

(k) CJCSM 3170.01B, *Operation of the Joint Capabilities Integration and Development System*, 11 May 2005.

(l) CJCSI 6212.01D, *Interoperability and Supportability of Information Technology and National Security Systems*, 8 March 2006.

(m) *Net-Centric Operations and Warfare Reference Model (NCOW RM) V1.0*, September 2003.

(n) *Net-Centric Checklist, V2.1.3*, Office of the Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer, 12 May 2004.

(o) *A Modular Open Systems Approach (MOSA) to Acquisition, Version 2.0*, September 2004.

(p) DoD IT Standards Registry (DISR), http://disronline.disa.mil.

(q) *Net-Centric Attributes List*, Office of the Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer, June 2004.

(r) *Global Information Grid (GIG) Key Interface Profiles (KIPs) Framework (DRAFT)*, Version 0.95, 7 October 2005.

1.2 Overview

Net-Centric Enterprise Solutions for Interoperability (NESI) provides, for all phases of the acquisition of net-centric solutions, actionable guidance that meets DoD Network-Centric Warfare goals. The guidance in NESI is derived from the higher level, more abstract concepts provided in various directives, policies and mandates such as the *Net-Centric Operations and Warfare Reference Model* (*NCOW RM*) and the ASD(NII) *Net-Centric Checklist*, references (m) and (n), respectively. As currently structured, NESI guidance is captured in documents covering architecture, design and implementation; a compliance checklist; and a collaboration environment that includes a repository of guidance statements and code examples.

More specifically, NESI is a body of architectural and engineering knowledge that guides the design, implementation, maintenance, evolution, and use of the Information Technology (IT) portion of net-centric solutions for military application. NESI provides specific technical recommendations that a DoD organization can use as references. Stated another way, NESI serves as a reference set of compliant instantiations of these directives.

NESI is derived from a studied examination of enterprise-level needs and, more importantly, from the collective practical experience of recent and on-going program-level implementations. It is based on today's technologies and probable near-term technology developments. It describes the practical experience of system developers within the context of a minimal top-down technical framework. Most, if not all, of the guidance in NESI is in line with commercial best practices in the area of enterprise computing.

NESI applies to all phases of the acquisition process as defined in references (a) and (b) and applies to both new and legacy programs. NESI provides explicit counsel for building in net-centricity from the ground up and for migrating legacy systems to greater degrees of net-centricity.

NESI subsumes a number of references and directives; in particular, the Air Force *C2 Enterprise Technical Reference Architecture (C2ERA)*[1] and the Navy *Reusable Applications Integration and Development Standards (RAPIDS)*.[2] Initial authority for NESI is per the Memorandum of Agreement between Commander, Space and Naval Warfare Systems Command (SPAWAR), Navy PEO C4I & Space and the United States Air Force Electronic Systems Center, dated 22 December 2003, Subject: Cooperation Agreement for Net-Centric Solutions for Interoperability (NESI). The Defense Information Systems Agency (DISA) formally joined the NESI effort in 2006.

---

[1] Air Force C2 Enterprise Technical Reference Architecture, v3.0-14, 1 December 2003.

[2] RAPIDS Reusable Application Integration and Development Standards, Navy PEO C4I & Space, December 2003 (DRAFT V1.5).

1.3 Releasability Statement

This document has been cleared for public release by competent authority in accordance with DoD Directive 5230.9 and is granted *Distribution Statement A: Approved for public release; distribution is unlimited*. Obtain electronic copies of this document at the following site: http://nesipublic.spawar.navy.mil.

1.4 Vendor Neutrality

The NESI documentation sometimes refers to specific vendors and their products in the context of examples and lists; however, NESI is vendor-neutral. Mentioning a vendor or product is not intended as an endorsement, nor is a lack of mention intended as a lack of endorsement.

Code examples typically use open-source products since NESI is built on the open-source philosophy. NESI accepts inputs from multiple sources so the examples tend to reflect whatever tools the contributor was using or knew best. However, the products described are not necessarily the best choice for every circumstance. Users are encouraged to analyze your specific project requirements and choose tools accordingly. There is no need to obtain, or ask contractors to obtain, the open-source tools that appear as examples in this guide. Similarly, any lists of products or vendors are intended only as references or starting points, and not as a list of recommended or mandated options.

1.5 Disclaimer

Every effort has been made to make NESI documentation as complete and accurate as possible. Even with frequent updates, this documentation may not always immediately reflect the latest technology or guidance.

1.6 Contributions and Comments

NESI is an open-source project that will involve the entire development community. Anyone is welcome to contribute comments, corrections, or relevant knowledge to the guides via the Change Request tab on the NESI Public site, http://nesipublic.spawar.navy.mil, or via the following email address: nesi@spawar.navy.mil.

1.7 Collaboration Site

The Navy has established a collaboration site to support NESI community interaction. It is located at https://nesi.spawar.navy.mil (user registration required). Use this site for collaborative software development across distributed teams.

2 Introduction

This document contains technical guidance for developing new systems and migrating current systems to conform to the ASD(NII) Net-Centric Checklist, reference (n), and the reference architecture described in the NESI document set (more specifically, Section 4.3 of Part 1 and various sections of Part 4). It provides enterprise design patterns for mission applications and services. This guidance will continue to evolve as our understanding of net-centricity evolves, and to reflect updates and changes to reference (n).

This section contains guidance for programs moving toward net-centricity and an overview of net-centric design tenets. See *NESI Part 1: Overview* for definitions of terminology related to net-centricity.

The remainder of this document lists design tenets in the areas of data, services, information assurance/security, and transport. The organization follows that of reference (n). The final two sections provide a service definition template and a set of net-centric standards.

2.1 Audience

The intended audience for this document includes the following:

- Program managers
- Deputy program managers
- Contracting officers
- Chief engineers
- Contractor personnel
- Enterprise and software architects

2.2 Net-Centric Design Tenets

Reference (n) contains a series of design tenets with a set of questions designed to gather system information. The design tenets help the DoD leadership understand how net-centricity is evolving.

This document organizes the reference (n) design tenets into the following sections.

- Data (Section 3)
- Services (Section 4)
- Information Assurance/Security (Section 5)
- Transport (Section 6)

Each design tenet provides specific technical guidance to enable the system to satisfy its net-centric requirements.

The technical guidance statements are written in a form suitable for inclusion in acquisition documents. It is not necessary to include every guidance statement. Instead, use these guidance statements as part of the overall system engineering analysis of a program to facilitate its evolution to net-centricity.

Not all design tenets can be satisfied by strictly technical guidance. All elements of Doctrine, Organization, Training, Materiel, Leadership, Personnel, and Facilities (DOTMLPF) must participate in the evolution of net-centricity.

3 Data

The *DoD Net-Centric Data Strategy*, reference (i), is a key enabler of DoD transformation. Significant attributes of the data strategy include the following:

- Ensuring that data are understandable and trustable, and that they are visible and accessible when and where needed to accelerate decision-making.
- "Tagging" data (intelligence, non-intelligence, raw, and processed) with metadata that supports discovery by both known and unanticipated users in the enterprise.
- Posting data to shared spaces that all users can access, except when limited by security, policy, or regulations.
- Posting in parallel with processing; Task/Post/Process/Use replaces the Task/Process/Exploit/Disseminate paradigm.
- Separating data from applications so that users may choose different applications to exploit the same data.
- Handling information only once to eliminate duplicate, non-authoritative data.

This section explains the design tenets surrounding data and data assets. A data asset is any entity that involves data. For example, a database is a data asset composed of data records.

3.1 Design Tenet: Make Data Visible

Data visibility requires an integrated environment of metadata models about the data assets. Perform forward and/or reverse engineering to capture metadata that describes the data assets of a node. Making data visible (even if not accessible) helps develop information about the node and its applications through insights such as the following:

- Essential missions that define the reason for the enterprise; the ultimate goals and objectives that measure enterprise accomplishment.
- Procedures performed by various groups in the enterprise that achieve these essential missions.
- The specific databases, information systems, and processes that groups use to accomplish aspects of the essential missions.
- Context-independent semantic templates of data elements and mechanisms for configuring into data models, as determined by subject matter experts.
- Mechanisms for configuring data models into databases used by organizations in the enterprise.

3.1.1 Guidance: General

Make all data assets visible, even if they are not accessible.

3.1.2 Guidance: DoD Discovery Metadata Specification

Use the DoD Discovery Metadata Specification (DDMS)[3] and all of its attributes to describe data assets.

3.1.3 Guidance: Metadata Generation

If possible, generate discovery metadata automatically.

3.2 Design Tenet: Make Data Accessible

Data accessibility requires defining data assets that exist within acceptable boundaries of security, along with the information necessary to access them. Relational databases automatically contain metadata about data assets. This guidance extends that definition to XML data that may exist independently or that are mapped to and/or from relational data.

3.2.1 Guidance: XML Requirement

Always use XML format to exchange information across systems. Define and implement an XML version of each external interface in all systems. If a system makes data available to external partners, that data must be available in the form of an XML document. This is required even if none of the current known partners want or send XML data.

Systems may implement other external data exchange mechanisms if an XML interface is supported. The use of other data-exchange mechanisms or other well-known document formats is contrary to the spirit of Web enablement and should be avoided in all but the small minority of cases where benefits outweigh costs.

3.2.2 Guidance: XML Interface Specification

The system that defines an XML interface shall do the following:

- Specify the syntax of the XML documents it accepts and produces.
- Use the XML Schema standard to express these specifications. Refer to *XML Schema Best Practices*[4] for guidance on creating XML schemas.
- Enter the schema in the *DoD Metadata Registry and Clearinghouse*.[5] This should occur as early as possible in the development process. Consult designated DoD XML Namespace Managers for guidance in choosing element, attribute, and type identifiers.

An XML interface is responsible for the following actions:

- Accept input data, produces output data, or both.
- Encode this data in XML documents.

---

[3] DoD Discovery Metadata Specification (DDMS), Version 1.3**,** 29 July 2005, Deputy Assistant Secretary of Defense, (Deputy Chief Information Officer).

[4] *XML Schema Best Practices*, http://www.xfront.org.

[5] *DoD Metadata Registry and Clearinghouse*, http://xml.dod.mil.

- Specify the schema of the XML documents it accepts and produces.
- Provide documentation that allows programmers and users to understand the meaning of those documents.
- Is implemented by a runtime service that accepts and produces such documents.

3.2.3 Guidance: XML Interface Usage

A system that uses an XML interface defined by some other system shall record this fact in the *DoD Metadata Registry and Clearinghouse*.

3.2.4 Guidance: XML Transport

Systems must implement one version of each XML interface that is accessible through a URL using HTTP/HTTPS. Systems may implement other versions of the interface using other transport mechanisms, such as FTP or SMTP, as long as they also support the HTTP version.

3.2.5 Guidance: Open-Standard Alternatives to XML Format

Information that is customarily exchanged using a well-known open-standard format does not have to be made available in XML. For example, systems may transfer image data in JPEG format, and email messages may continue to use RFC822 headers. it is not necessary to develop an equivalent XML interface for these.

A list of the exception formats will be made available.

Information intended for presentation that is currently held in Standard Generalized Markup Language (SGML) format does not have to be immediately converted into XML. However, systems should consider future migration from SGML to XML.

3.2.6 Guidance: Proprietary Alternatives to XML Format

Information that can only be expressed using closed proprietary formats does not have to be made available in XML. For example, systems may continue to exchange word processor files in Microsoft® Word (DOC format). It is not necessary to develop an equivalent XML interface for this information.

3.3 Design Tenet: Make Data Understandable

The following actions help make data understandable:

- Start with well-defined data ontologies, taxonomies, and vocabularies.
- Develop standard data elements from these according to ISO/IEC Standard 11179.[6]
- Employ these data elements as the basis for data model structure templates.
- Employ the templates throughout the database models and operating databases.

[6] http://www.iso.org.

This ensures that standard data elements are the semantic basis for all database models. This guidance extends to the semantics of XML schemas that may exist independently or that are generated from database data models.

3.3.1 Guidance: XML Schema Usage

Systems shall specify XML schemas according to the following rules:

- Developers shall search the *DoD Metadata Registry and Clearinghouse* for existing XML schemas that can be reused in the system interfaces. When existing XML schemas are reused, this fact shall be recorded in the *DoD Metadata Registry and Clearinghouse*.
- If an existing XML schema is close to but not exactly what was specified, assess whether the differences are significant enough to justify developing a new schema. If there is no measurable operational impact, consider changing the system requirements to use the existing schema. If developing a new schema, ensure that it is easy to translate between the new schema and the closely related, existing schema.
- Developers shall review proposed XML definitions with the designated DoD XML Namespace Manager for their COI.
- A system shall not unilaterally define XML schemas for information that it does not produce.
- Developers should look for government and industry consortia that produce XML definitions. Some of these may be suitable for reuse in the system interfaces.
- Systems should define XML interfaces in collaboration with their known information exchange partners.

3.3.2 Guidance: XML Schema Documentation

A system that defines an XML interface shall provide adequate documentation for the meaning of the documents it produces or accepts. This documentation shall be expressed as annotations on the XML schema for the interface.

The system must supply a text definition for every element, attribute, and enumeration value defined in the schema. Refer to the XML Schema specification[7] for more information on schema annotations.

As a next level of documentation, the system should be able to provide a "metadata story" for each XML element with information from the metadata represented by the view, physical, logical, conceptual, and data element models.

3.4 Design Tenet: Make Data Trustable

The key to trusting data is to know that when accessing the same data from any location, the data are either identical or reconcilable. This requires a clear understanding of the database

[7] World Wide Web Consortium (W3C), http://www.w3c.org/XML.

architecture classes. Formalize and enforce authoritative data sources, which must be as current as possible and distributed in a timely manner.

3.4.1 Guidance: General

Systems shall use the Resource Descriptors and Security Descriptors specified by the *DoD Metadata Registry and Clearinghouse* to provide data validity and security information.

3.4.2 Guidance: Authoritative source

Systems shall identify the authoritative source for each data element. This identification shall include the purpose.

3.4.3 Guidance: Aggregated Data

Aggregated data can often exceed the security level of the individual data elements. Systems shall recognize and account for the possibility of an increased security level when aggregating data.

3.5 Design Tenet: Make Data Interoperable

Data assets that have been properly analyzed and stored are interoperable. This analysis and storage includes information such as names, XML tags, data types, lengths, precision, scale, restricted value domains, and definition metadata. ANSI-standard SQL database management systems provide tools to wrap data in XML tags. This allows the software agents to mine the XML tags.

3.5.1 Guidance: XML Wrapped Data

If XML wrapped data are intended for exchange, configure them in terms of standard transactions with headers, trailers, and bodies.

3.5.2 Guidance: XML Schema Validation

Systems that produce XML documents shall guarantee that the XML documents are valid according to the XML schema they have published in the *DoD Metadata Registry and Clearinghouse*. Systems that receive XML documents should validate them against the schemas published by the source system.

3.6 Design Tenet: Provide Data Management

3.6.1 Guidance: General

Systems shall provide a process to define, develop, and maintain an ontology (e.g., schemas, thesauruses, vocabularies, keyword lists, and taxonomies) to adequately support and improve all design tenets.

3.7 Design Tenet: Be Responsive to User Needs

3.7.1 Guidance: General

The system shall include users in the COI data specification process.

4 Services

A **service** is a contractually defined behavior provided by a software component through a service interface. Services enable the rapid development and deployment of capabilities that can be combined with other services to provide a range of simple and complex functions.

Services have the following advantages:

- They are self-contained, software building blocks that are URI addressable, reusable, and easily distributed.
- They are loosely coupled from clients, reducing integration costs.
- They expose capabilities independent of their implementation.
- They insulate users from implementation and data changes.

4.1 Design Tenet: Service-Oriented Architecture (SOA)

In a **Service-Oriented Architecture** (SOA), mission capabilities are provided as services that one or more clients may invoke. This promotes flexibility and reuse, and enables complex software systems to be composed from stable interfaces.

In a service-oriented operational environment, from a service consumer's perspective, a service should be a "black box." It provides a specified function or functions at a specified level of performance for an agreed-upon cost.

An SOA is a design style for building flexible, adaptable, distributed-computing environments. Service-oriented design is fundamentally about sharing and reusing functionality across diverse applications. Service-oriented design is based on the following best practices:

- Design the application and system functionality as accessible and reusable services.
- Expose service functionality through programmatic interfaces.
- Maintain an abstraction layer between service interfaces and service implementations.
- Describe service interfaces using standard metadata.
- Advertise and discover services using standard service registries.
- Communicate with services using standard protocols.

Implementing an SOA assumes a robust IP-based network—including ground-based, maritime, airborne, and space-based systems—that provides reliable communications for IP-enabled systems. This guidance does not address issues related to providing robust communications or network quality of service. For more about these issues, refer to the GIG ES and NCES specifications.

4.1.1 Guidance: Service-Oriented Architecture

Build services in accordance with the technical standards and conformance requirements prescribed by the current version of the WS-I Basic Profile.[8] Check with potential vendors for their level of compliance.

Developers should do the following:

- Use the WS-I Sample Application as a model for implementing and documenting services.
- Use test tools authorized by WS-I that verify conformance with the current version of the WS-I Basic Profile.
- Build and develop security extensions as prescribed in the current version of the WS-I Basic Security Profile.[9]

4.1.2 Guidance: Service Description

Describe services using a standard **Service Definition Framework** (SDF). Section 7, *Service Definition Framework Template* provides a detailed specification for service definition and implementation.[10] The SDF provides service users, customers, developers, providers, and managers with a common frame of reference. Its structure and methodology enable full definition of the **Service Access Points** (SAPs) for the service. The purpose of the SDF is *not* to describe the internal workings of a service. Rather, it concentrates on defining the boundary conditions for accessing a service through its service access point. The SDF also includes specific technical parameters and engineering-level data that prospective service developers and providers can use to design and implement new enterprise service offerings.

An SDF entry will be completed for each enterprise service. Each service will subsequently be registered in a service registry (e.g., the NCES Service Discovery service or Air Force Service Management Tool). The SDF provides the basis for a design specification where potential implementers of a new service will find the information required to implement the service. The recommendation is that the SDF address the following information for each service:

- What the service does
- How the service works (from a black box perspective)
- Any required security mechanisms or restrictions
- Any pertinent performance or quality of service (Q0S) information
- Points of contact for the service:
    - Who is providing the service
    - Who is responsible for the daily operation of the service

---

[8] Web Services Interoperability (WS-I) Basic Profile specification, (http://www.ws-i.org).

[9] Web Services Interoperability (WS-I) Basic Security Profile, (http://www.ws-i.org).

[10] The NESI SDF element specification is consistent with draft DoD policy outlined in Draft NCIDS Global Information Grid Net-Centric Implementation Document—Service Definition Framework (S300), 21 December 2005 (v2.0; based on the original NESI SDF).

    - Who is developing the service
- The specifics of how to bind to (access or use) the service.

SDF entries for each enterprise service are expected to be registered in at lest one service registry.

In a future release, NESI will provide a set of usage scenarios. The proposed service development lifecycle is included in this release to assist service implementers n the development and maintenance of an SDF entry during the lifecycle of an enterprise service. Scenarios include the following:

- Creating an SDF Entry
- Changing a Registered SDF Entry
- Deprecating a Registered SDF Entry
- Accessing a Registered SDF Entry

The proposed SDF Lifecycle is consistent with the DoD Acquisition Steps defined in the DoD 5000 series Directives and Instructions. The table in Section 4.1.2.2 below describes the proposed steps for the SDF lifecycle, along with associated business processes, the service owner and mandatory categories for each phase.[11]

4.1.2.1 Service Profiles

A service profile captures the black box architecture of a service. It would precede and guide one or more service implementations documented in association with the SDF. The use of a service profile becomes critical in the case of those Enterprise Services that have more than one implementation and implementer across the enterprise. The profile provides the guidance needed to ensure that multiple service implementations provide a common consumer interface and are interoperable.

4.1.2.2 Proposed SDF Lifecycle

Lifecycle Element	Description	Business Processes	Service Owner	Mandatory Categories by Phase
Concept Development	Identify possible need for a new service and create justification for service	Examine mission threads and search for services to fulfill them. Identify capability gaps. These gaps become services within classification domains. Create high level business or mission capability statement. Perform initial cost analysis and Analysis of Alternatives. Define acquisition approach and organizations to execute following phase	Portfolio manager	Service name, service description, schedule

[11] Draft NCIDS Global Information Grid Net-Centric Implementation Document—Service Definition Framework (S300), 21 December 2005 (v2.0; based on the original NESI SDF).

Lifecycle Element	Description	Business Processes	Service Owner	Mandatory Categories by Phase
Requirements and Architecture	Define service architecture and requirements	Identify specific organizations for each type of user, Define service requirements and semantics. Define service architecture to include interaction with other services and systems, basic service capabilities and service deployment approach. Perform Systems Program Office (SPO) level cost analysis.	Portfolio manager to acquirer	Semantic model, pedigree, information security marking, contacts (PoC)
Service Design	Create service "black box" interface specs for handoff to developers	Start configuration management - finalize semantics, point to metadata repository - finalize classification details - determine service level agreements (SLAs) offered - finish WSDL	Acquirer	Operations, number of operations, security mechanisms, access criteria and restrictions, service level spec, network requirements, SAP
Service Build	Develop/purchase service	Development - generally follows contractors best practices	Acquirer	Consumer patterns, schedule Beta, operational reference
Service Testing	Assure service meets specifications and requirements	Acceptance test - meets specs - play well with others - interoperability - "seals of approval" - which authoritative bodies	Acquirer to operator/ sustainer	Schedule-integration,
Service Deployment	Install service instance(s)	Configuration management: -updating humans/summary from monitoring and measuring – coarse-grained triggers for action (scaling)	Operator/ sustainer	Schedule-deployment
Service Operation	Operate service; concludes with EOL announcement.	Configuration management: -updating humans/summary from monitoring and measuring - coarse grained triggers for action (scaling)	Operator/ sustainer	Schedule-operation
Service Deprecation	Service is still being operated but is to be replaced or retired; concludes with service EOL	Work with consumers to adopt new version of service, or replacement service(s) as appropriate	Operator/ sustainer	Schedule-deprecation
Service Retired	Service is not operating; sservice definition information is still available for use/reuse; concludes with purging of service definition information	Service migration and reuse	Sustainer	Schedule-retire

4.1.2.3 Notional SDF Concept of Operations

The *Notional SDF Concept of Operations* outlines a theoretical ConOps for Service Discovery. ConOps for SDF is focused on why a service is needed and how it is used. The ConOps addresses the following issues:

- **Key Assumptions**:
  - Location, composition, extensibility, syntax, failover, information assurance, alignment to COIs and applicable security classification level
  - Governance
  - Services are made available via an Enterprise Service Bus or via the Web services stack
  - The SDF will be used for defining services from many sources and multiple languages
- **Creation of an SDF Entry**
  - Two scenarios in which a service will require the creation of an SDF entry:
    - Capability already exists and will be "service enabled"
    - Capability does not exist
  - The SDF entry becomes part of the Key Interface Profile (KIP) for that service
- **Services Lifecycle and SDF Development Process Flow**
  - Establishment of a business case
    - Warfighter or COI has defined a need
    - Service requirements analysis and definition
    - Funding
    - Resources assigned
  - Design
  - Development
  - Test
  - Deploy
- **SDF Implementation**
  - SOA
    - Publishing
    - Discovery
    - Binding
  - Operations and maintenance
    - Change Management
    - Deprecation
    - Monitoring and maintenance

  Under SDF Implementation, NESI also advises that ConOps include Portfolio Management and Capability Planning. NESI will add these components in future versions.

4.1.2.4 SDF Guidance and Best Practices

- Describe all services using a standard Service Definition Framework (SDF).
  - Adhere to DoD Policy as a core definition for the SDF

    - Extensions can be made to core definition to suit specific needs
- May want to extend "Required" fields (from core SDF)
- Capture and track associated Lifecycle Phase
- The "Owner" of the service (and SDF) will change as the Lifecycle Phase changes; need to update the SDF at each Lifecycle phase.
- Begin capturing SDF data at the earliest Lifecycle Phase as possible, preferably at the Concept Development phase.
    - Not all information will be available
    - Recommended to trace service capability back to operational needs, shortfalls and requirements
- Make SDF data accessible by storing contents either in an XML document in conformance with the XML Schema, or it can be in the form of a set of database tables with a front-end.
    - The XML Schema or database tables will contain all elements and attributes of the core (and extended) SDF
    - Best Practices – database tables with a front-end:
        - Group SDF data elements into logical categories and reflect such in the User Interface (UI) for ease of use; do not just provide one large input form.
        - Reports are high value; being able to view SDF data via reports allows for relationships to be discovered and services to be managed (Portfolio Management, Capability Based Planning).
        - Role-based access for data editing is vital for information assurance and integrity; don't want Service Owner A to edit Service Owner B's SDF
        - Enforce security policies at the Data Level rather than at the application and/or UI level; provides stronger information assurance and accountability (audits); allows data entries and data fields to be customized to each user/role.
- Capture SDF data from discrete choices (lists) rather than just "free text"; while free text can be searched via key word, it does not allow as much capability for data relationships and data mining.
- Make SDF data understandable and use terminology/labels relevant to the particular domain (enterprise).
- Designate minimally required data with respect to appropriate Lifecycle Phase needed for a complete understanding of the service at that phase.
    - Recommend that "Required" fields be tied to Lifecycle Phase; some information may not be available at earlier phases, but would be required before eventually moving into a later phase.

4.1.3 Guidance: Service Access Point (SAP)

The system shall describe the services it provides through SAPs. From a service provider perspective, SAPs can be abstracted away from the back-end or internal processing activities of the service. Looser coupling between SAP and service internals enables a service provider to change the internal workings of the back end, such as moving to a new version of a database, without changing the SAP.

4.1.4 Guidance: Service Design

Design services around operational requirements and service consumers' needs. Base the service specifications on the needs of the initial users, since it is impossible to know all the possible service consumers. Provide an extensible interface so the service design can support future needs.

4.1.5 Guidance: Service Design Characteristics

Design services in accordance with best practices and patterns. For example, a service design should specify the information objects that are communicated across its interface in terms of enterprise metadata (e.g., time, location). These enable semantic agreement between the information objects.

Design information objects to minimize the number of transactions across the service interface. An example of this is a request for an ATO, possibly constrained by a time and location attribute, followed by a reply containing the ATO that is applicable to a specific area of interest and time.

NESI Part 5: Developer Guidance contains a robust set of design patterns for services that capture best practices.

4.1.6 Guidance: Service Implementation Characteristics

Services must do the following:

- Document the open standards used.
- Use vendor- and platform-independent messages.
- Identify addresses using a Universal Resource Identifier[12] (URI).
- Use defined and documented service interfaces. Register the interface description in XML using the *DoD Metadata Registry and Clearinghouse*. Describe service interfaces using Web Services Description Language (WSDL).[13]
- Pass enterprise or COI objects, defined by their respective metadata, across its service interface.
- Use extensible service interfaces with versioning, independent of the interface implementation version.

Implementation information focuses on the technical implementation details that prospective service developers or providers need to design new services, or a service that uses another service. These attributes typically include items like the WSDL description of the service, details of a service's API interface point, and a description of service dependencies.

4.1.7 Guidance: Service Level Agreement (SLA)

Services must have a documented Service Level Agreement (SLA) that shall do the following:

---

[12] A URL is a special case of a URI.

[13] Web Services Description Language (WSDL) 1.1, http://www.w3.org/2002/ws/desc, http://www.w3.org/TR/wsdl.

- Include quantitative measures for service usage, performance analysis, continuity of operations plan, and performance across the range of bandwidths provided by the node.
- Have terms that can be monitored and managed by the node's management services.
- Define responsibility for day-to-day service operations and procedures for reporting problems.

The service shall specify whether its SLA will be managed at the enterprise level as well.

4.1.8 Guidance: Service Interfaces

Interface information includes descriptions of service features, service functionality, service provider identification, instructions on how to access and use the service through the SAP, and so on. The interface information should also discuss the different form factors that a service supports, such as a PDA.

Express the service interfaces in WSDL in accordance with the current version of the WS-I Basic Profile.

Services must not implement XML-RPC.

Services communicate using XML-based messages. Format service messages using the Simple Object Access Protocol (SOAP) in accordance with the current version of the WS-I Basic Profile. SOAP provides a structured message format and XML schema for communicating XML information.[14]

4.1.9 Guidance: Node Responsibilities for Services

The node infrastructure should enable mission application software to be instantiated as services; this includes software libraries that support SOAP and WSDL processing. Node responsibilities include the following:

- Using Web services standards (SOAP and WSDL) to interoperate applications across nodes.
- Providing secure access to components in accordance with node and GIG IA/Security policies and services.
- Designing services to be managed by the node in accordance with enterprise policy. Management services will typically be part of the node component framework environment (e.g., J2EE application server, .NET management environment) that is used in conjunction with NCES Enterprise Service Management.
- Providing the capability to name and register components for local use within the node (e.g., JNDI). Component registration mechanisms shall interface or extend to service registration mechanisms, such as registration in the NCES Discovery service. If the component is only visible to the local node, it does not have to be registered in the NCES Discovery service.

[14] Simple Object Access Protocol (SOAP) 1.1, (http://www.w3.org/TR/SOAP).

4.1.10 Guidance: Service Registration

The system shall register services using the standard service metadata in a directory available to the nodes in the enterprise. This directory may be based in the node, an NCES Discovery Service, or both. At a minimum, the system shall identify a service by a Universal Resource Identifier (URI).

The node shall register services as resources with the NCES Policy Management Service and control access to services using the NCES Policy Decision Services. The NCES Resource Attribute Services must provide access to service attributes.

4.1.11 Guidance: Service Security

Service security requirements follow:

- Use security mechanisms provided by the node. These must include mutual authentication over an encrypted channel such as SSL, authorization, confidentiality, integrity, and non-repudiation.
- Services must support role-based access control (RBAC) mechanisms.
- Nodes shall provide interfaces to NCES security services.
- Nodes shall establish trust relationships with other nodes in the enterprise using the NCES Domain Federation Services.

Security information provides detailed information about the security specifications of the service, such as restrictions on who can use or access the service. For example, they would indicate if the user must present a valid DoD PKI certificate to access the service.

Security extensions, including transferring security assertions or tokens, and measures to implement integrity and confidentiality shall use WS-Security for SOAP Message Security[15] in accordance with the current version of the WS-I Basic Security Profile.

4.1.12 Guidance: Support for Service Orchestration

Nodes shall provide the capability to compose mission capabilities from one or more services using a service orchestration or workflow mechanism based on industry standards such as BPEL.

4.2 Design Tenet: Open Architecture

4.2.1 Guidance: General

Mission application software shall be separable from the supporting node and shall access the node through public interfaces. A public interface is based on public standards governed by a recognized standards organization (e.g., IEEE, W3C, OASIS) and does not support proprietary extensions.

[15] Web Services Security (WSS) SOAP Message Security 1.0 (WS-Security 2004), OASIS Standard 200401, March 2004 (http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0)

4.2.2 Guidance: Component Based

Architect mission application software as components integrated within a node. The node shall provide run-time and resource management services (e.g., component management, security, virtual machines, memory management, object management, resource pooling).

Nodes shall include component frameworks based on commercially available solutions[16] without proprietary extensions. Use of extensions should be wrapped via the appropriate design pattern.

Architect and manage mission application software that spans multiple nodes in a manner that aligns with all of the supporting nodes.

4.2.3 Guidance: Public interfaces

The node shall provide the mechanism for components to expose public interfaces. The interface must be separate from the implementation. Base the public interface mechanism on the node component framework.[17] These public interfaces must be visible to other components in the node.

4.2.4 Guidance: Layered Software Architecture

Layer application software using an N-tier architecture. At minimum, use discrete client, presentation, middle, and data tiers.

4.2.5 Guidance: Client Tier

The client tier supports a wide range of device types such as desktop computers, laptops, mobile, wireless, and PDA devices. It supports direct interaction with the user.

4.2.6 Guidance: Presentation Tier

The presentation tier provides presentation content to a range of client device types supported by the node (e.g., HTML, XML, WML). Implement presentation components with the mechanisms in the node's component framework.[18]

4.2.7 Guidance: Middle Tier

The middle tier supports the construction of componentized business logic and public interfaces (e.g., interface classes). Base business components on programming mechanisms provided by the component framework chosen by the node (e.g., Enterprise Java Beans, CORBA® services, COM components). Specific business logic elements, such as data validation, may reside in other tiers.

---

[16] Examples include Java Enterprise Edition (Java EE), Common Object Request Broker Architecture (CORBA®), Microsoft® .NET Framework, and OMG Data Distribution System (DDS).

[17] Examples include Enterprise Java Bean interface, CORBA® IDL interface, and Microsoft® .NET interface.

[18] Examples include Java Server Pages (JSPs), Java servlets, Active Server Pages (ASPs), static HTML pages, and dynamic HTML pages.

4.2.8 Guidance: Data Tier

Base access to the data tier within nodes on industry open-standard mechanisms such as SQL or JDBC/ODBC. Use services to access data across nodes. This should be consistent with the current version of references (h) and (i).

4.2.9 Guidance: Wrapping Legacy Systems

Wrap legacy application software with an interface that is accessible from the node; for example, use Java Connector Architecture on a Java EE platform. See *NESI Part 3: Migration Guidance* for details.

4.3 Design Tenet: Scalability

4.3.1 Guidance: General

Design services and components to use resource management mechanisms provided by the Node Platform Infrastructure (NPI) that enable scalability under load. For example, use buffer and connection pools, tuned to the expected user load, to enable concurrent user sessions with acceptable performance.

4.4 Design Tenet: Availability

4.4.1 Guidance: General

Design services and components to meet the availability requirements of the node. The implementation should use the node maintenance strategies and management mechanisms provided by the NPI.

4.5 Design Tenet: Reliability

4.5.1 Guidance: General

TBD

4.6 Design Tenet: Flexibility

4.6.1 Guidance: General

TBD

4.7 Design Tenet: Accommodate Heterogeneity

4.7.1 Guidance: Service Structure

Be able to deploy services separately from the supporting node. The services should access the node through public interfaces.

4.7.2 Guidance: Service Configuration

Be able to configure services on each node on which they are deployed. Use external configuration file mechanisms (e.g., deployment descriptors for Java EE applications) to specify the configuration. Do not use hard-coded configuration parameters that require a binary tool to update or that require a recompile and relink.

4.7.3 Guidance: Node Structure

Nodes provide the infrastructure and rules for assembling, configuring, deploying, securing, operating, and managing mission applications and services. For more information, see *NESI Part 4: Node Guidance.*

Nodes are responsible for provisioning their diverse mission application and services. They must configure and operate them in accordance with enterprise management policy and NCES Enterprise Service Management services.

4.8 Design Tenet: Decentralized Operations and Management

4.8.1 Guidance: General

Services should provide a management interface that can be accessed either by the node's management services or the NCES Enterprise Service Management services.

4.9 Design Tenet: Enterprise Service Management

4.9.1 Guidance: Service Management

Nodes manage services; this includes their assembly, deployment, fault isolation, and run-time monitoring. To do this, nodes use available management services, either NCES Enterprise Service Management services or local management services. Services must expose a management interface that the node management services can access.

4.9.2 Guidance: Provisioning of Enterprise Services

If an enterprise service is available from DoD/DISA as an NCES capability, the node must enable its applications and components to access it.

If a required NCES service is not available, then the node may implement the service locally based on technical standards provided by DoD/DISA (when available).

If there is no DoD/DISA standard describing the NCES service, the node shall choose standards based on best commercial practice.

In all cases, node and application developers must maintain a separable service implementation. This enables them to replace local node implementations if and when the NCES service becomes available.

The node must be able to access the following categories of NCES:

- Application
- Collaboration
- Discovery
- Enterprise service management
- Information assurance/security
- Mediation
- Messaging
- Storage
- User assistant

5 Information Assurance/Security

Information assurance (IA) refers to measures that protect and defend information and information systems. Their goal is to ensure confidentiality, integrity, availability, and accountability. To this end, they provide capabilities to protect against attack, detect attacks, monitor attacks, and react to attacks.

Many of the existing solutions to IA problems (and many of the requirements in existing IA regulations) assume that both clients and servers are located on the same physical or logical network. They rely heavily on perimeter or boundary protection. The interoperability and loose coupling requirements of an SOA make those security models far from adequate.

In an SOA, the boundaries are not clearly defined. Services may be exposed to external clients and not bound to a physical location. The client and service providers may be governed by different security policies.

A net-centric IA strategy needs to be based on a service-level view of security rather than on perimeter security. Developing new security models is necessary to determine how to establish the necessary trust relationships between service requestors and service providers, and to select the most adequate and appropriate authentication and authorization mechanisms. To implement a net-centric IA strategy, programs must provide the following:

- Integrated identity management, permissions management, and digital rights management
- Adequate confidentiality, availability, and integrity

The guidance for each of the security design tenets expands on those differences and suggests ways to comply with the requirements and development characteristics of net-centric systems. Much of this guidance assumes that the appropriate enterprise services are in place. Therefore, we also address some of the intermediate steps that help you migrate to a full service-oriented architecture. We place special emphasis on standards and mechanisms that support XML and SOAP, such as WS-Security. WS-Security is the OASIS standard that establishes interoperability mechanisms to exchange security tokens or assertions. It enhances the confidentiality and integrity of SOAP messages.

This guidance supports but does not provide all the guidance required to comply with DoD PKI Policy.[19]

5.1 Design Tenet: Net-Centric IA Posture and Continuity of Operations

This tenet refers to the assignment of Mission Assurance Category (MAC) and Confidentiality Level to a given application, node, or system. The MAC reflects the importance of information relative to the achievement of DoD goals and objectives, particularly the warfighter's combat

[19] Department of Defense Instruction 8520.2, *Public Key Infrastructure (PKI)* and *Public Key (PK) Enabling*,1 April 2004.

mission. Mission Assurance Categories primarily determine the requirements for availability and integrity.

There are three defined mission assurance categories:

- MAC I for systems with vital operational needs
- MAC II for systems that are important to deployed or contingency forces
- MAC III for systems supporting day-to-day businesses that do not materially affect support to deployed forces

The complete definitions for those categories are included in DoDD 8500.1.[20] The security requirement for each combination of mission assurance category and its confidentiality level are in DoDI 8500.2.[21]

5.1.1 Guidance: Mission Assurance Category

In a net-centric environment, the MAC must consider not just the intrinsic properties of the node or service, but also its impact on other Information Operations that may call upon it.

When developing a node or service, account for its potential use by other missions and adjust the MAC appropriately. Incorporate adequate protection and integrity requirements into the design that are commensurate with those potential uses.

Typically, not all of the potential uses of a node or service are known up front. Therefore, developers must make assumptions about how critical missions may use the node or service when they determine requirements. It may be necessary to modify the MAC to accommodate future, critical missions.

5.2 Design Tenet: Identity Management, Authentication, and Privileges

Authentication mechanisms are based on credentials presented by the requestor. Those credentials may be something the user knows (e.g., passwords), something the user is (e.g., biometrics), something the user has (e.g., smart card), or any combination.

Each approach is associated with the strength of an authentication. The weakest methods are password-based and the strongest are combinations of biometrics and smart cards.

There are also different strengths within each method. For instance, systems that require complex passwords are stronger than those that accept simple ones, and systems using retina or fingerprint readers are stronger than those that use finger length.

---

[20] Department of Defense Directive 8500.1, *Information Assurance (IA)*, 24 October 2002 (Certified Current as of 21 November 2003).

[21] Department of Defense Instruction 8500.2, *Information Assurance (IA) Implementation*, 6 February 2003.

5.2.1 Guidance: User Authentication

In an SOA, the user must normally be authenticated at the "edge" application or node, or at the very first network access.

Applications should be ready to accept strong authentication methods as early as possible. If possible, migrate authentication tasks to an authentication server and make applications rely on tokens or assertions from the server to authenticate users.

5.2.2 Guidance: Identity Management

Use authentication assertions to propagate identities in a secure and trusted way throughout the enterprise. Those assertions must indicate not just the identity, roles, or attributes of the requestor, but the strength of the mechanism used to ascertain its identity.

Generate a Trust Model to specify the proper trust relationships and the path for authentication assertions. For closed community configurations, these schemes may involve the use of a Kerberos-type single sign-on device.

5.2.3 Guidance: Multi-Tier Authentication

While considering the specific method used and its relative strength, remember that SOA service providers may require stronger authentication than that invoked by the service requestor. In those cases, a multi-tier authentication may be required, re-authenticating the original user with the server by transferring appropriate credentials.

To avoid future multi-tier authentication problems, use strong authentication methods such as PKI certificates whenever possible.

5.2.4 Guidance: Authorization Processes

Access authorizations are determined by identification parameters and by the nature and contents of the request. Make the authorization decision at the access boundary, isolating the application from changes in policy and authorization technology. For example, in Java EE applications, base the access control decisions on the deployment descriptor.

Use node-managed security (sometimes referred to as declarative or container-managed security), unless application requirements require programmatic authorizations, where individual actions within the service are authorized based on the nature or parameters of the request. This is sometimes referred to as **programmatic security**.

5.2.5 Guidance: Role-Based Authorizations

The user's role is the best way to establish access authorizations, since it isolates the service provider from changes in the user population. In the Role-Based Access Control (RBAC) environment, the access decision is based only on the role parameters. Roles should be supplied by a trusted entity in association with the user identity. Roles should never be supplied directly by the user. Base all access authorizations on roles as early as possible.

As systems and applications evolve toward SOAs, do not require global or enterprise-wide names for all roles. Allow different environments to select their own role names. This implies that a request to a different environment may have to map the requestor role name to one of the

role names accepted by the service provider. Furthermore, the service provider may use programmatic authorizations that require a finer granularity of role definitions. When using RBAC, specify the level or granularity of the roles and their names with the service description. Detailed guidance on how to apply RBAC in a net-centric environment will need to be based on new enterprise services supporting role name management.

When you expand the role characteristics to include many user attributes (e.g., level of clearance, level of training, specific assignment location), the role-based access controls are sometimes referred to as attribute-based access controls.

Additional parameters may factor in an authorization decision. They are related to the security context of the request. For example, the privileges may be different if the same user accesses the service from within a local secure boundary, from a remote location, or through a VPN from a home environment. Until enterprise guidance based on these parameters exists, the system designer must implement local solutions based on node guidance.

When the application retrieves access control information from an external policy decision point, or retrieves policies for its own resources, it should do so with eXtensible Access Control Markup Language (XACML). XACML supports exchange of access control information using XML.

5.2.6 Guidance: Validation of Authentication Information

A service provider may receive requests that include the original authentication information from the requestor. DoD uses Public Key Infrastructure (PKI) certificates for authentication information.[22] The most effective way for the provider to ascertain the validity of the authentication information is to confirm it through a PKI mechanism.

When the requestor identification information is received through a security assertion, the service provider must authenticate that the assertion has been validated by an entity that the provider trusts. This can also be accomplished through PKI signatures. The signatures must encompass and link both the assertion and the actual request. The user must describe whether PKI is used, the complete scheme of how the certificates will be verified, the timeliness of the requests, and the current validity of the credential (i.e., verification that the certificates have not been revoked). All these tasks require extensive PKI infrastructure support, which is still under development.

Systems should migrate to PKI authentication as it become available, and start using it as a baseline to provide enterprise authentication services.

5.3 Design Tenet: Mediate Security Assertions

One of the main issues in an SOA is how to convey user authentication and access authorization to a service provider, and how to demonstrate that the information is valid and authentic. Use security assertions or security tokens. These are statements generated or validated by an entity that the service provider trusts. The WS-Security standard allows you to attach those assertions or tokens to SOAP messages.

[22] http://iase.disa.mil/.

5.3.1 Guidance: Security Assertions

The main goal is to transfer assertions using an XML-based standard such as the Security Assertion Markup Language (SAML).

For close community configurations, start with Kerberos security tokens. Establish implicit trust relationships between entities to circumvent formal validations through the use of trusted channels (e.g., SSL transfers).

Transfer security tokens or security assertions using the general purpose mechanism provided for associating security tokens or assertions with SOAP message contents as specified in the WS-Security Standard. Kerberos and other tokens shall use the Binary Security Token provision. Use SAML assertions in the context of WS-Security as specified in the upcoming WS-Security SAML Token Profile.[23]

5.3.2 Guidance: Chained Requests

When requests need to be chained (i.e., forwarded to third parties), the security assertions must cover the origin and destination, all intermediate assertions, and the required chain of trust. Earlier request implementations may separate a chained request into separate transactions.

5.4 Design Tenet: Cross-Security-Domains Exchange

Exchange information across security boundaries using air-gap interfaces, electronically enforced one-way interfaces, content-based encryption, content-sensitive security guards, multilevel trusted databases, and multilevel systems. The data exchange may be from low to high or high to low. In an NCW environment, many of the service requests and their corresponding trust assertions may have to cross security boundaries; that is, they must originate and terminate at entities with different security classification levels.

5.4.1 Guidance: Cross-Domain Services

In a net-centric environment, enterprise-wide services are the most efficient way to handle those transactions and implement cross-domain solutions. Special cross-domain services should be developed to provide validated resources capable of transferring information between security domains operating at different security classifications. To support net-centric warfare effectively, cross-domain solutions must transition from current models to an agile and flexible; robust and available; trusted, yet economical, solution set. The most effective method is to provide those services at the enterprise level, compatible with the GIG and NCES services. As enterprise-wide cross-domain solutions are implemented, additional guidance will have to be developed.

NESI architecture should fully consider and incorporate the capabilities and procedures of centralized cross-domain solutions as they are being developed. If possible, systems should demonstrate an evolution towards these enterprise-wide solutions. Current systems must still rely on existing secure guards solutions or one-way solutions.

---

[23] Web Services Security: SAML Token Profile, Working Draft 15, 19 July 2004 (http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-1.0).

5.5 Design Tenet: Encryption and HAIPE[24]

Enterprise services must enable secure transmission of identification and role assertions through the use of trusted paths. A **trusted path** is a communications path with the following characteristics:

- There is reasonable confidence that there has not been any malicious alteration of the information.
- The data are timely, meaning they originated within a small preceding period of time.

Note that the definition of "timely" is not the same for all types of information systems. Services should specify an appropriate definition based on the type of information system (e.g., event-driven, transaction-based) and the type of security threat (e.g., replay attack).

5.5.1 Guidance: Trusted Paths Establishment

Several communications protocols can accomplish those tasks in a TCP/IP environment. The system must use either SSL, IPSec, or HAIPE protocols, and eventually incorporate message-level XML encryption.

5.6 Design Tenet: Employment of Wireless Technologies

5.6.1 Guidance: Wireless Technologies

All data transmissions need integrity assurances that the information has not been altered. For transmission of sensitive or classified information, there should also be confidentiality assurances that the information has not been exposed to unauthorized users. In the case of wireless technologies, consider those assurances in the context of lack of finite boundaries for information protection, and the possibilities of spoofing (i.e., unauthorized insertions of information). Many standards are being developed for the protection of wireless networks using cryptographic means

Systems should encrypt all traffic when using wireless technologies, using established standards.

5.7 Other Design Tenets

Although service-oriented architectures (SOAs) require new concepts from perimeter-based models, the equipment that provides the services will still be located in finite clusters. Providing boundary or perimeter protection may result in additional assurances against penetration from non-DoD external links. The main defense security regulations, namely DoD 8500 Series and DCID 6/3,[25] apply to SOA components. Some of the regulations may not directly apply, or they may require special considerations when applied to SOAs.

---

[24] High Assurance Internet Protocol Encryptor.

[25] Director of Central Intelligence Directive 6/3, *Protecting Sensitive Compartmented Information Within Information Systems*, 5 June 1999.

This section contains information on the following topics:

- Integrity and confidentiality
- Boundary protection
- Intrusion detection
- Auditing

5.7.1 Guidance: Integrity and Confidentiality

Encrypt requests and responses to achieve the appropriate level of confidentiality protection. Nodes must use protocols such as the following:

- Secure Socket Layer (SSL) or Transport Level Security (TLS) for transport layer security
- IPSEC for network layer
- Secure MIME (S/MIME) for email traffic

Eventually, encryption mechanisms must migrate to message-level security mechanisms. These mechanisms use XML-encryption and message integrity protection based on XML-Digital Signature (XML-DSIG). Use those standards in the context of enhancements to SOAP messages, as specified in the WS-I standard Web Services Security (WS-Security) and the WS-I Basic Security Profile.

Secure messages must also include security timestamps to prevent recording and playback of valid messages. All timestamps must use Coordinated Universal Time (UTC), also referred to as Greenwich Mean Time (GMT) or Zulu (Z) time.

5.7.2 Guidance: Firewall Configurations

Continue using firewalls and proxy servers to protect the physical boundary of clusters of equipment supporting SOAs. Carefully define the "boundary" of the system. The firewall must prevent unauthorized penetrations, but it must be carefully programmed to reduce the inherent additional risks of SOAs.

Those risks come from allowing inbound HTTP/HTTPS access to the Web server applications. An example of such a threat would be an ill-intended SOAP message that causes internal application buffer overflow while looking completely benign to the firewall and Web server.

Use new XML-capable firewalls as they become available.

5.7.3 Guidance: Intrusion Detection Systems

Use adequate monitoring to determine anomalies or failures that can impair mission performance. Intrusion detection systems should detect unauthorized access and penetration attempts. Use detection and protection mechanisms to automatically detect and prevent illicit actions, and complement them with manual reporting of anomalies or specially detected events. Enable automatic reconfiguration or recovery features only for limited and well-defined conditions.

5.7.4 Guidance: Intrusion Reporting

In an SOA, there must be some centralization of automated reports to establish enterprise security awareness. These reports are coupled with correlation and analysis of events detected at multiple nodes. The scope of the environment conducting the correlation depends on the availability of software agents in individual nodes and the availability of resources that can establish the correlation of events. The scope may range from a few systems at a given location to all activities within a theater of operations. An even broader analysis may occur through manual reporting at an enterprise-wide level.

5.7.5 Guidance: Audit Events Linkage

Configure and use individual system audit mechanisms. For SOAs, audits should be complemented by mechanisms that correlate events in different nodes and provide network-wide forensics. Time stamping and logging of all inter-node messages help link events and actions involving multiple nodes. All time stamping must use UTC.

5.7.6 Guidance: Use of Audits for Attribution

Use proper logging and request auditing to satisfy attribution requirements (i.e., determination of the individual responsible for the action). This should occur at both the requestor and service provider sites. Logging must use UTC.

5.7.7 Guidance: GIG Policy Compliance

All systems must be developed in accordance with the IA requirements contained in DoDI 8500.2 for the appropriate Mission Assurance Category and Sensitivity Level. If applicable, they must also be compliant with DCID 6/3. Systems must leverage the guidance and technologies described in DoD CIO *Guidance and Policy Memorandum 6-8510* "DoD GIG Information Assurance" of 16 June 2000 and the *End-to-End Information Assurance* of the GIG, Version 1.0, 30 June 2004 and its future versions.

5.7.8 Guidance: Certification and Accreditation

All systems must be certified and accredited in accordance with the DoD Information Technology Security Certification and Accreditation Process (DITSCAP) as documented in DoDI 5200.40 (December 30, 1997) and DoDM 8510-1M (July 2000). In addition, Air Force systems should comply with the certification and accreditation section in AFI 33-202 "Network and Computing Security," 26 September 2003.

6 Transport

The **Transport Infrastructure** is a foundation for net-centric transformation in DoD. To realize the vision of a Global Information Grid, ASD(NII)/DoD CIO has called for a dependable, reliable, and ubiquitous network that eliminates stovepipes and responds to the dynamics of the operational scenario. To construct the Transport Infrastructure, DoD will do the following:

- Follow the internet model.
- Create the GIG from smaller component building blocks.
- Design with interoperability, flexibility to evolve, and simplicity in mind.
- Provide a common, black-core IP network for both unclassified and encrypted classified information.

This guidance is aimed at users and providers of transport services. Where DoD-adopted standards are required, reference (p) is the primary source reference unless otherwise specified. Many of the technology and implementation specifics associated with these tenets are still in the development stage and have not yet reached the maturity of guidance. In addition, the *Checklist* tenets do not cover every capability needed to implement or use a net-centric transport infrastructure. Therefore, satisfying this guidance is only a beginning for transport users and providers. Many aspects of the tenets and guidance are still being developed and resolved.

Unless otherwise specified, the guidance in this section is based on evolving DoD guidance.

6.1 Design Tenet: IPv6

In the next four to five years, the adoption of IPv6 throughout the DoD and in the Federal Agencies will pass a major implementation threshold. Most DoD bases and other facilities will be IPv6 capable. Most of the key components of the technology are in place for native deployment of IPv6 or dual existence of IPv4 and IPv6.

A 9 June 2003 ASD(NII)/DoD CIO memo, "Internet Protocol Version 6 (IPv6)" is the first in a series of memos[26] addressing DoD transition to IPv6. The main points of the directives follow:

- The tentative original goal for IPv6 transition completion is set for FY08.
- Enterprise-wide deployment of IPv6 is conducted by DoD in controlled, integrated and cohesive manner,[27] for which three major Milestone Objectives (MOs) have been outlined. Currently only those systems that have been approved as MO1 pilots are allowed to switch to IPv6 in operational environments.

[26] 9 June 2003, "Internet Protocol Version 6 (IPv6)"; 29 September 2003, "Internet Protocol Version 6 Interim Transition Guidance"; 28 November 2003, "Internet Protocol Version 6 Transition Plan Coordination and Interim Tasking";16 August 2005, "DoD Internet Protocol Version 6 Policy Update"; 16 August 2005, "DoD Internet Protocol Version 6 Pilot Nominations"

[27] March 2005, "The Department of Defense (DoD) Internet Protocol Version 6 (IPv6) Transition Plan

- The DoD IPv6 Transition Office has been established within DISA and given responsibility for coordination of the transition efforts, providing required infrastructure, and insuring that unified solutions are used across DoD. Each service has its Transition Office responsible for providing technical guidance and transition governance to programs. This includes transition plans development (to be coordinated into a master plan by DISA), dispensation of IP addresses originating from DISA, waiver policy implementation, MO1 pilot nominations, etc.
- A mandate, to minimize costs of transition, is that all GIG assets being developed, procured or acquired must be IPv6 capable (in addition to maintaining interoperability with IPv4 capabilities). The DoD CIO directives contain an outline for the "IPv6 capable" requirement, while a detailed specification is still being developed.
- The transition to IPv6 should be accomplished through the normal technical refresh cycle whenever possible.

6.1.1 Guidance: Support IPv6 Transition

6.1.1.1 Transport Service Users

Transport service users must be able to interoperate with interfacing transport service providers who use either IPv6 or IPv4 during the transition from IPv4. New applications should be IP version agnostic and shall employ an operating system that supports both IPv4 and IPv6 (**dual-stacked**). For existing IPv4 service users, migration plans should be developed and approved by the governing authority (e.g., Component IPv6 Transition Office).

6.1.1.2 Transport Service Providers

Transport service providers interfacing with non-transitioned networks must support both IPv6 and IPv4 during the transition from IPv4. Mechanisms proposed to allow the two protocols to coexist and inter-operate during the transition phase from IPv4 to IPv6 include the following:

- Incorporating both IPv4 and IPv6 support in routers and computers; this is called **dual stacking**. This is a preferred way to ensure the interoperability between systems during the transition period.
- Transporting IPv6 traffic through IPv4 networks by encapsulating IPv6 packet in IPv4 and vice-versa; this is called **tunneling.** During the initial enabling of IPv6 in operational environments (MO1[26,27]) in controlled enclaves, tunneling becomes a useful communication mechanism between the enclaves. Tunneling should be considered only as a temporary solution.
- Placing translation gateways between IPv4 and IPv6 networks or hosts. This is the only mechanism allowing a native IPv4-only device to communicate with IPv6-only device. The expectation is that these devices will not be needed until the later stages of transition for dominant IPv6 devices to communicate with some lingering native IPv4 legacy devices.[28]

[28] David Green and Bob Grillo, SRI International,"The State of IPv6 A Department of Defense Prospective February 2005" – prepared for the DoD IPv6 Standards Working Group

In all cases, IPv6 transport provider planning must be coordinated with the service IPv6 Transition Office.

6.1.2 Guidance: Support IPv6 IP Security Features

6.1.2.1 Transport Service Providers

A goal of transport service providers is to support IPv6 IP security features for data integrity and confidentiality.

IPv6 provides improved security features in comparison to IPv4 through IPSec and mandatory support for end-to-end security. The Service Transition Office should be able to provide guidance on utilizing any of the IPv6 security features in the context of the service enterprise transition plan.

6.1.3 Guidance: Implement DoD-Adopted IPv6 Standards and Products

6.1.3.1 Transport Service Providers

Transport service providers shall implement DoD-adopted IPv6 standards and products. The list of standards directly relevant to DoD and approved for the use on DoD networks is maintained in the DISR, reference (p).

6.2 Design Tenet: Packet Switched Infrastructure

The GIG network comprises a number of component networks. Each component network must pass data both internally among its network members and externally to or from other GIG component systems. As such, the internet model that applies to the development of the GIG transport infrastructure must be designed as an IP datagram delivery system. The delivery system consists of a packet-switched communications facility in which a number of distinguishable component networks (including any networks external to this system) are connected together using routers.

We need to refine technologies such as routing standards and QoS mechanisms to achieve the end-to-end functionality required by the GIG. These should be designed and applied within the framework of a packet-switched transport infrastructure. Many infrastructure functional elements are required to implement a packet-switched infrastructure.

6.2.1 Guidance: Implement Interface to One and Only One Network Layer (Layer-3) Protocol for Datagrams

6.2.1.1 Transport Service Users

Transport service users shall implement interface(s) to *one and only one* network layer (Layer-3) protocol for datagrams. This applies to datagrams passed both within a component network and to those destined for external networks.

6.2.1.2 Transport Service Providers

Transport service providers shall implement *one and only one* network layer (Layer-3) protocol for datagrams passed both within a component network as well as those destined for external networks. This paradigm may be violated during transition periods, but the fundamental goal remains a single inter-network protocol.

GIG component system designers should consider how the component transport infrastructure will accept externally-generated IP datagrams that are destined for hosts inside their system. This allows their system to "attach" to the GIG. They should also consider how their component infrastructure will deliver internally generated IP datagrams to hosts outside their system, and how it will serve as a transit network for externally generated IP datagrams.

6.3 Design Tenet: Layering and Modularity

Change is probably the only inviolable characteristic of the commercial internet model. Moreover, change occurs at different rates in different elements of the network/protocol stack. Design the GIG transport infrastructure to accommodate that change. The most effective way to allow differential change in a system is through modular, layered design.

Although market forces and commercial practice have deprecated the ISO-OSI model, it still provides excellent guidelines for implementing a layered design. These guidelines still apply to the development of the GIG transport infrastructure.

In a layered design, each layer is independent and adds value to the set of services offered by lower layers. The services provided to/from a layer are well defined; however, the precise approach for providing these services is *not* specified. ISO defined a number of principles to consider when developing a layered design and applied those principles to develop the seven-layer OSI architecture.

While a seven-layer approach may not be the solution for the GIG transport infrastructure, GIG component system designers should consider the principles ISO defined to facilitate interoperability and to reduce technology interdependencies that add to system complexity. A subset of these principles that apply to the GIG transport infrastructure is provided below.

6.3.1 Guidance: Define Layer Boundaries and Interfaces

6.3.1.1 Transport Service Users

Transport service users shall implement one or more interfaces to the defined transport service delivery point(s), where the services description can minimize the number of interactions across the interface boundary(ies). The transport service provider networks should provide the interface boundary definition(s). The goal is to minimize the cross-layer physical and functional interdependencies to facilitate GIG transport infrastructure growth and interoperability.

6.3.1.2 Transport Service Providers

Transport service provider networks shall provide one or more interface boundaries as the transport service delivery point(s), where the services description can minimize the number of interactions across the boundary(ies). To the maximum extent possible, functionality

implemented within each OSI layer of the transport service implementation should only interface with the adjacent upper and lower layers via defined interfaces. The goal is to minimize the cross-layer physical and functional interdependencies to facilitate GIG transport infrastructure growth and interoperability.

6.3.2 Guidance: Ensure Functions are Modular and Separable

6.3.2.1 Transport Service Users

Transport service users shall create a layer of easily localized functions. These functions should enable developers to totally redesign the transport services and its protocols to take advantage of new advances in architectural, hardware, or software technology without changing the services and interfaces with the adjacent layers.

6.3.2.2 Transport Service Providers

Transport service provider networks shall create a layer of easily localized functions. These functions should enable developers to totally redesign the layer and its protocols to take advantage of new advances in architectural, hardware, or software technology without changing the services and interfaces with the adjacent layers.

Transport service providers shall identify all instances in their transport infrastructure where a logical or physical coupling or dependency exists between different layers of the protocol stack. The goal is to minimize the cross-layer physical and functional interdependencies to facilitate GIG transport infrastructure growth and interoperability.

6.3.3 Guidance: Minimize Complexity of Layered Implementation

6.3.3.1 Transport Service Providers

Transport service provider networks shall keep the number of layers small enough to reduce the complexity of describing, integrating, and maintaining the layers.

6.4 Design Tenet: Transport Goal

A design goal of the GIG is network convergence. Voice, video, and other multimedia traffic should be packetized and transported along with data traffic over a common IP network. Another transport goal is the convergence of encrypted classified information flows on a common black IP network. This corresponds to the direction of commercial industry, where telecommunications providers and corporate telephony are migrating to IP.

A primary benefit of convergence is that it eliminates the expensive hardware and complexity of separate, dedicated networks that support serial-based traffic (voice and video teleconferencing). Other benefits include greater efficiency of bandwidth and the ability to introduce new features based on converged services.

6.4.1 Guidance: Support Interfaces with Converged Traffic Networks

6.4.1.1 Transport Service Users

Transport service users shall implement interfaces to, or transition to, a transport infrastructure supporting full convergence of traffic on a single IP inter-network, using DoD-adopted standards and DISA/JITC-certified (voice) solution sets.

Transport service users shall identify and minimize all instances where performance standards cannot be met using a converged transport infrastructure (e.g., where dedicated, single-traffic-type transport service is required). The goal is to minimize cross-layer physical and functional interdependencies to facilitate GIG transport infrastructure growth and interoperability.

6.4.1.2 Transport Service Providers

Transport service providers shall implement and/or plan to support full convergence of traffic on a single IP inter network using DoD-adopted standards.

Voice, video, and other multimedia traffic have relatively strict delivery requirements with regard to latency and jitter. This requires networks to support the QoS features identified in Section 6.7, Design Tenet: Differentiated Management of Quality-of-Service.

The DoD-adopted set of standards appears in reference (p). These include standards for Voice over IP (VoIP) and video teleconferencing (VTC) based on ITU H.323.

Voice over IP (VoIP) refers to a set of standards and technologies that allow voice to be transmitted over IP networks. The industry has embraced two different sets of standards:

- ITU H.323 is the more mature and complete set of standards, which encapsulates ISDN call signaling over an IP-based network.
- A more recent set of standards, developed by the IETF, is based on the Session Initiation Protocol (SIP). The SIP standard concerns simple call placement and is designed to be easily expandable.

Since there are currently two options for VoIP, the DoD plans to select a set of mandated standards within reference (p).

Video teleconferencing over IP is based on ITU H.323. This is an umbrella standard of ITU recommendations that address audio, video, signaling, and control for packet-switched networks.

6.5 Design Tenet: Network Connectivity

Network connectivity shall be provided to all end points, such as wide- and local-area networks, and direct connections to mobile end users. This guidance addresses the Layer-2 or terminal-to-network interfaces.

6.5.1 Guidance: Manage Scalability and Complexity

6.5.1.1 Transport Service Users

Transport service users shall quantitatively evaluate scalability before formulating a final design. The evaluation should identify any transport infrastructure design drivers regarding the number

of hosts that need to be supported and/or number of networks that are required to support the technologies chosen for the specific transport service use.

6.5.1.2 Transport Service Providers

Transport service providers shall quantitatively evaluate scalability before formulating a final design. The evaluation should identify any limitations regarding the number of hosts/network or number of networks that could be supported using the technologies chosen for the specific transport infrastructure design.

One way to reduce complexity is to use a minimal set of standards/protocols in developing the GIG transport infrastructure. This implies that any selected standard/protocol has the capacity to serve as large a percentage of the GIG as possible. Component systems of the GIG should select standards/protocols that can scale to the enterprise. GIG component system designers should evaluate their transport infrastructure design to identify any instances where different technology/protocols perform the same function (e.g., internal routing).

6.5.2 Guidance: Optimize Use of COTS Products

6.5.2.1 Transport Service Users

Transport service users shall use open, COTS products as much as possible.

GOTS and/or vendor-unique products lead to interoperability and evolvability issues. Use them only when there is an overarching, unique, DoD requirement driving that selection. Any protocols, standards, etc., that are not included in reference (p) and/or could not be purchased off-the-shelf from a commercial networking vendor shall be documented and justified against the resulting impact to GIG component system interoperability.

6.5.2.2 Transport Service Providers

Transport service providers shall use open, COTS products as much as possible.

GOTS and/or vendor-unique products lead to interoperability and evolvability issues. Use them only when there is an overarching, unique, DoD transport infrastructure requirement driving that selection. Any waveforms, protocols, standards, etc. that are not included in reference (h) and/or could not be purchased "off the shelf" from a commercial networking vendor shall be documented and justified against the resulting impact to GIG component system interoperability.

6.6 Design Tenet: Concurrent Transport of Information Flows

This tenet addresses the use of **Inline Network Encryptors (INEs)** that allow all security domains to be globally “known” to the Layer-3 encrypted backbone network. This is a fundamental shift from current link-by-link encryption. Utilizing a black-core network should provide a significantly streamlined communications infrastructure that also makes more efficient use of the available bandwidth through the invocation of QoS/CoS based IP datagram multiplexing.

High Assurance Internet Protocol Encryptor (HAIPE) devices are among the critical technologies that should enable the black-core IP-network vision to become a reality. However, a

number of technical challenges must be solved before the vision can be realized across all functional domains and COIs. These include the following:

- Support for IP-based QoS/CoS
- Support for dynamic unicast IP routing
- Support for dynamic multicast IP routing
- Support for mobility
- Support for simultaneous IPv6 and IPv4 operation

6.6.1 Guidance: Implement INE Standards and Products to Support Traffic Convergence

GOTS and/or vendor-unique products lead to interoperability and evolvability issues. Use them only when there is an overarching, unique, DoD requirement driving that selection.

6.6.1.1 Transport Service Providers

Transport service providers shall implement DoD-adopted INE standards and products, when available, to support traffic convergence from multiple security domains on a single IP inter-network. Currently, DoD is engaged in IETF-standards working groups and vendor communities to accelerate development of new standards in the areas of security, tactical communications, QoS, and reliable networking. Some standards have been adopted for QoS and HAIPE. A product list is being developed for infrastructure, hardware, software, and other categories of IPv6 products.

6.6.2 Guidance: Document Approach to Information Infrastructure with Black Core

GOTS and/or vendor-unique products lead to interoperability and evolvability issues. Use them only when there is an overarching, unique, DoD requirement driving that selection.

6.6.2.1 Transport Service Providers

Transport service providers shall document their approach to providing an information infrastructure with a black core.

6.7 Design Tenet: Differentiated Management of Quality-of-Service

Some applications in the GIG require firm service guarantees, while others operate correctly if they receive services that are differentiated with respect to one or more performance characteristics.

Differentiated Services or **DiffServ**[29] aggregates flows into coarse classes and then treats the packets in these classes differentially. Due to this aggregation, and the resulting absence of a need to consider individual flows beyond the edges of an internet, DiffServ exhibits good scaling properties. However, in the absence of additional mechanisms, DiffServ provides only preferential, differentiated levels of service and not guarantees.

Various approaches are being explored, with none yet adopted. DoD QoS/CoS Working Group is investigating complete end-to-end QoS frameworks providing both differentiated and guaranteed QoS. They are developing a DoD roadmap and baseline architecture strawman. The architecture needs to define transport user and transport provider functions, such as where packets are labeled (application or router with Service Level Agreement).

6.7.1 Guidance: Support Quality of Service (QoS) and Class of Service (CoS)

6.7.1.1 Transport Service Users

Transport service users shall interoperate with interfacing transport service providers who use standardized DoD QoS/CoS in accordance with the DoD QoS/CoS Roadmap. As the interfacing networks are transitioned to standardized QoS/CoS, transport service users should plan to migrate to maintain interoperability.

6.7.1.2 Transport Service Providers

Transport service providers shall prioritize traffic based on class of user, application, or mission. Lower priority data flows should be preempted if a higher priority flow is initiated and insufficient resources exist to carry both flows simultaneously. This capability, referred to as Class of Service (CoS) support, corresponds approximately to the notion of Multi-Level Priority and Preemption (MLPP). The GIG and its components must support both QoS and CoS in accordance with the DoD QoS/CoS Roadmap and policies.

6.8 Design Tenet: Inter-Network Connectivity

A fundamental tenet of the commercial internet model is that the complexity of the internet belongs at the edges. Certain required end-to-end functions can only be performed correctly by the end systems themselves. Any network, however carefully designed, will be subject to failures of transmission at some statistically determined rate.

The best way to cope with this is to accept it and give responsibility for the integrity of communication to the end systems. This principle drives the complexity of the network to the edge and limits state information held inside the network. This increases the robustness of end-to-end communications since application state can now only be destroyed by a failure of the end systems.

Many issues need to be resolved to mature the guidance for this tenet, especially for transport users whose data traverse different media with different performance characteristics. In some situations it may not be desirable to follow this design tenet.

---

[29] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, *An Architecture for Differentiated Services*, RFC 2475, IETF, December 1998.

For example, the use of TCP proxies, which may be required to achieve adequate performance across satellite assets, runs counter to this tenet. The proxy (part of the network and not an end system) maintains state information on the TCP session between two end-user systems, but it cannot guarantee that the function that is being performed by TCP is being accomplished.

Avoid implementing "intelligence" within the network whenever possible.

6.8.1 Guidance: Support Internetwork Connectivity Using DoD-Adopted Standards

6.8.1.1 Transport Service Providers

Transport service providers shall support inter-network connectivity using DoD-adopted standard protocols contained in reference (p), such as BGP4. Any protocols or standards that are not included in reference (p), such as performance-enhancing proxies, should be documented and justified against the resulting impact to GIG component system interoperability.

6.9 Design Tenet: DoD IT Standards Registry (DISR)[30]

6.9.1 Guidance: Justify and Document All Standards that Are Not Included in the DISR.

6.9.1.1 Transport Service Users

Transport service users must justify and document all standards that are not included in reference (p), especially those that impact transport service infrastructure design.

6.9.1.2 Transport Service Providers

Transport service providers must justify and document all transport standards that are not included in reference (p).

6.10 Design Tenet: RF Acquisition

6.10.1 Guidance: Justify, Document, and Obtain a Waiver for All Radio Terminal Acquisitions that Are Not JTRS/SCA Compliant

6.10.1.1 Transport Service Users

Transport service users must justify, document, and obtain a waiver for all radio terminal acquisitions that are not JTRS[31]/SCA[32] compliant.

---

[30] See reference (p).

[31] JTRS: ASD(NII)/DoD CIO Memo, Subject: Radio Frequency (RF) Equipment Acquisition Policy, 17 June 2003.

[32] SCA: http://jtrs.army.mil.

6.10.1.2 Transport Service Providers

Transport service providers must acquire JTRS/SCA-compliant radio terminals and coordinate with OSD and the JTRS JPO.

6.10.2 Guidance: Minimize RF Bandwidth Requirements

Use appropriate transmit protocols, compression standards, and other techniques when interfacing RF networks to the GIG environment. The RF environment with its much more constrained and error prone propagation environment requires techniques that minimize bandwidth requirements.

6.11 Design Tenet: Joint Net-Centric Capabilities

ASD(NII)/DoD CIO[33] issued a memorandum that identifies a number of key C4ISR programs (see Table 1) for integrating into the GIG.

Table 1 – Programs Identified in ASD(NII)/DoD CIO Memorandum Subject: Joint Net-Centric Capabilities, July 15, 2003

• All Space Terminal Acquisition • All Intelligence, Surveillance, and Reconnaissance (ISR) programs • Teleports • Warfighter Information Network-Tactical (WIN-T)	• All radio and data link applications • Global Command & Control System (GCCS), Joint and Service Variants • Crypto Modernization • Distributed Common Ground/Surface Systems (DCGS) • Other programs as noted	• All C2 programs • Deployable Joint Command & Control (DJC2) • High Assurance Internet Protocol Encryption (HAIPE) • Future Combat System (FCS) • Programs under the FORCEnet umbrella

The memo highlights programs that are required to develop transition plans for integrating transport components with the following GIG joint net-centric capabilities:

- Internet Protocol version 6 (see Section 6.1, *Design Tenet: IPv6,* for guidance details)
- Net-Centric Enterprise Services[34]
- JTRS/SCA (see Section 6.10, *Design Tenet: RF Acquisition,* for guidance details)
- Global Information Grid Bandwidth Expansion (GIG-BE)[35]

---

[33] Assistant Secretary of Defense for Networks and Information Integration, Memorandum: Joint Net-Centric Capabilities, 15 July 2003.

[34] Defense Information Systems Agency, Net-Centric Enterprise Services (NCES) Program Management Office, http://www.disa.mil/main/nces.html

- Transformational Communications Satellite/Advanced Wideband System[36]
- End-to-end information assurance (see Section 5, Information Assurance/Security, for guidance details)

Reference (n) also highlights the need for the programs listed in Table 1 to include in transition plans, use of guard technologies, and standards and protocols for connectivity with allied and coalition partners.

6.11.1 Guidance: Employ NCOW RM

6.11.1.1 Transport Service Users

Transport service users must use reference (m) to guide implementation of Joint Net-Centric capabilities. The reference model provides context for the types of architectures and computing infrastructures that the GIG transport systems and management functions must support. This enables common interfaces to transport-service-delivery points and interoperable-employment-of-transport services.

6.11.1.2 Transport Service Providers

Transport service providers must use reference (m) to define their architectures and guide implementation of Joint Network Centric capabilities. The GIG NetOps Architecture from GIG Version 1.0 was a central component used to develop reference (m). The reference model provides context for the types of architectures and computing infrastructures that the GIG transport systems and management functions must support. This enables common interfaces to transport-service-delivery points and interoperable employment-of-transport services.

6.12 Design Tenet: Operations and Management of Transport and Services

This tenet encompasses three equally important principles of NetOps and associated guidance:

- Develop manageable systems
- Use non-proprietary implementations
- Use accepted industry standards

NetOps:

- Is a coordinated, comprehensive set of operational concepts and structure that "fuses" Systems and Network Management, Information Assurance/Computer Network Defense, and Content Staging/ Information Dissemination Management into a single integrated operational construct.

---

[35] GIG-BE: ASD(NII)/DoD CIO Memo, Subject: Acquisition Decision Memorandum (ADM) – GIG Bandwidth Expansion (GIG-BE) Program, 3 January 2003.

[36] TSAT/AWS: Capabilities Development Document for the Transformational Satellite Communications (SATCOM) (TSAT) System, Draft Version 2, HQ AFSPC, Increment 1, 6 June 2003.

- Is an end-to-end capability that represents the integrated doctrine, force structure, and tactics, techniques, and procedures (TTP) needed to manage and direct the net-centric operations of the GIG.
- Encompasses all activities directly associated with the net-centric management and protection of GIG computing (including applications and systems), communications, and information assurance assets across the continuum of military operations.
- Actively integrates those capabilities with the goal of end-to-end, assured network availability, information delivery, and information protection.

6.12.1 Guidance: Develop Manageable Systems

Transport communications and network systems, services, sub-systems, sub-services, components, devices, and elements must be built from the ground up to be "manageable." They should also have the appropriate functional management capabilities.

In addition, transport communications and network services and systems should be proactively managed and operated to specific levels of service. These service levels are documented and published in Operational or Service Level Agreements (OLA/SLAs).

Management solutions for transport systems and services must be fully integrated with management solutions to ensure that the GIG is holistically operated and managed to support operational warfighter requirements. Operational management solutions should fully address all specific management functional areas; e.g., fault, configuration, accounting, performance, and security management.

6.12.2 Guidance: Use Non-Proprietary Implementations

Operational management capabilities and solutions should be based on non-proprietary implementations of industry accepted standards. An example is SNMP for IP-based networks.

Critical transport systems, subsystems, component, and elements shall be able to securely monitor, detect changes in, and report the following:

- Basic up/down operational status
- Performance information
- Operational configuration
- Security status

Management interfaces should be non-proprietary. They must be accessible to a wide variety of management products and solutions via open-standards-based interfaces. The interfaces should not require hard-coding to obtain operational status information about a particular system.

6.12.2.1 Transport Service Providers

To support the development of NetOps Situational Awareness capabilities, transport service providers must ensure that their operational management solutions can share operational status and other types of management information with management solutions operated by other types of service providers. The exchange must use non-proprietary standards-based interfaces.

While this could be as simple as offering a browser-accessible web interface using HTTP or HTTPS, management product vendors are beginning to implement web services interfaces that use SOAP to share information between management systems.

6.12.3 Guidance: Use Accepted Industry Standards and Emerging NetOps Concepts

Operational concepts, architectures, processes, and procedures used by transport communications and network providers must incorporate emerging NetOps concepts. They should be based on accepted industry standards.

6.12.3.1 Transport Service Providers

Transport service providers must take an active role in the growing NetOps community. They should help develop the operational policies, processes, and procedures that enhance the flow of information between different management domains. This will ensure that problems are proactively detected, isolated, and resolved with minimum impact to the user.

To support this goal, transport service providers should adopt and implement operational policies, processes, and procedures based on internationally accepted *de facto* Telecommunication Service Provider and IT Service Management (ITSM) standards.[37]

6.12.4 Guidance: Support Standardized DoD Service-Oriented Environment

6.12.4.1 Transport Service Users

Transport service users shall employ DoD-adopted standards for using transport infrastructure in the GIG-ES Enterprise Service Management (ESM)/NetOps service-oriented environment, rather than a domain or system-oriented environment.

6.12.4.2 Transport Service Providers

Transport service providers shall employ DoD-adopted standards for implementing transport infrastructure in the GIG-ES Enterprise Service Management (ESM)/NetOps service-oriented environment, rather than a domain or system-oriented environment.

A Working Group was established early in CY2003 to help develop DoD-level policy for operating in a service-oriented environment. Co-chaired by ASD(NII)/DoD CIO and DISA, this group has enjoyed wide participation and representation from across the Services as well as from key enterprise programs. The main focus of this group has been to formulate initial ESM/NetOps requirements for GIG-ES and for the Net-Centric Enterprise Services (NCES) Program. The group also identified DoD-level policy areas that may need to be revised to support net-centric operations in an SOA. In addition, the group has collaborated with the NetOps CONOPS group to broaden the current transport- and network-centric approach to one that is more holistic and consistent in monitoring, managing, and controlling systems, services, and applications, in addition to transport systems and networks.

---

[37] The TeleManagement Forum's Enhanced Telecom Operations Map™ (eTOM) and the Information Technology Infrastructure Library (ITIL®)

6.12.5 Guidance: Employ DoD-Adopted Standards to Support Cross-System and Domain Management

6.12.5.1 Transport Service Providers

Transport service providers shall employ DoD-adopted standards for operating and managing transport services. This includes interaction with counterparts in other networks or management domains, such as system or application managers.

Transport service providers shall specify interfaces and/or standards for the following:

- Sharing operational status and performance information
- Collecting and disseminating service management information
- Selecting the format in which it is made available (e.g., SNMP, XML, CIM, SOAP)

SNMP and XML are identified as mandated standards in reference (p), Volume I. CIM is an emerging standard identified in reference (p). CIM is also identified as a target standard in reference (m).

6.12.6 Guidance: Plan for Coalition Interoperability

6.12.6.1 Transport Service Providers

Transport service providers shall plan for operations and management of transport services. This includes interacting with counterparts in other networks or management domains used by coalition partners.

Most recent conflicts have involved not only U.S. forces, but forces from allies and coalition partners. In the future, U.S. information and communications systems must support interoperability with these groups.

This interoperability can be achieved in a variety of ways:

- Acquisition of common systems
- Development of diverse but interoperable systems
- Adherence to standards and commercial best practices

7 Service Definition Framework Template

This section provides the logical model to help the service implementer to understand the *big picture* for the Service Definition Framework (SDF). The logical SDF model, summarized in Table 2 below, provides the primary service element categories, and service element name and a brief definition. Each service element represents information that may or may not be relevant to the particular service being described. Some service elements may only be applicable during certain phases in the service lifecycle. Other service elements may not apply to specific technologies.

The attributes of a service that are necessary to effectively define and describe the service are identified within the SDF and organized into the following categories:

- Interface information
- Security information
- Service level information
- Implementation information
- Point of contract (POC) information
- Service Access Point (SAP) information

All categories, with the exception of the SAP, are abstract and allow the service to be defined so as to encourage semantic understanding of the service. The last category (SAP) is the concrete portion that is filled in after the service has been implemented and deployed. The SAP binds the abstract service specification to the concrete service interface as implemented by an actual process. Specific syntax, protocols and IP address required to use the functionality provided by the service are contained in the SAP.

In the table, the service elements have an associated cardinality for inclusion in the SDF. Cardinality is interpreted as follows:

- Cardinality = 1: Element is mandatory, one instance only
- Cardinality = 1..n: Element is mandatory, one to many ("n" = no upper limit, or upper limit is specified)
- Cardinality = 0..1: Element is optional, but limited to one instance if it is present
- Cardinality = 0..n: Element is optional, and there may be one instance or more if it is present.

Table 2 has an additional column, which is the recommended lifecycle phase where the given service element applies. A detailed specification of Service *"Data"* Elements will be included in a future release of NESI.

Table 2: SDF Service Element Development Lifecycle Phase

Service Element Category	Service Element	Cardinality	Service Development Lifecycle Phase
Interface information	ServiceName	1	Concept Development
	Service Description	1	Concept Development
	Semantic Model	0..1	Requirements & Architecture
	NumberOfDataTypes	1	Service Design
	DataTypes	0..n	Service Design
	NumberOfOperations	1	Service Design
	Operations	1..n	Service Design
	ServicePedigree	1	Requirements & Architecture
Security information	SecurityMechanisms	1	Service Design
	AccessCriteriaAndRestrictions	1	Service Design
	InformationSecurityMarking	1	Requirements & Architecture
Service level information	NumberOfServiceLevels	1	Service Design
	ServiceLevelSpecifications	0..n	Service Design
	NetworkRequirements	0..1	Service Design
Implementation information	ConsumerPatterns	0..1	Service Build
	NumberOfScheduleDates	1	Concept Development
	Schedule	1..n	Concept Development
	NumberOfOperationalReferences	1	Service Build
	OperationalReference	0..n	Service Build
	VersioningApproach	0..1	Service Design
POC information	NumberOfContacts	1	Requirements & Architecture
	Contacts	1..n	Requirements & Architecture
SAP information	NumberOfSAPs	1	Service Design
	ServiceAccessPoint	0..n	Service Design

8 Net-Centric Checklist Standards

The following partial list of standards represents the "backplane" set of standards that every system must follow, as documented in reference (n).

8.1 Web Foundational

- *Hypertext Transfer Protocol (HTTP) Version 1.1, IETF RFC 2616.*
- *Hypertext Markup Language (HTML) 4.01, W3C Recommendation.*
- *File Transfer Protocol (FTP), IETF Standard 9, IETF RFC 959.*
- *User Datagram Protocol (UDP), IETF Standard 6, IETF RFC 768.*
- *Transport Control Protocol (TCP), IETF Standard 7, IETF RFC 793.*
- *Internet Protocol (IP), IETF Standard 5, IETF RFCs 791, 792, 950, 919, 922, 1112.*
- *Simple Mail Transfer Protocol (SMTP), IETF RFCs 1870, 282*1.
- *Multi-purpose Internet Mail Extensions (MIME), IETF RFCs 2045-2049.*
- *Uniform Resource Locator (URL), Uniform Resource Identifier (URI), IETF RFCs 1738, 1808, 1866.*
- *Unicode universal character set, International Organization for Standardization (ISO) 10646, Universal Multiple-Octet Coded Character Set (UCS), IETF RFC 2277* http://unicode.org.

8.2 Web Emerging Standards or Best Practices

- HTTP State Management Mechanism, IETF RFC 2965, XML Schema 1.0 (http://www.w3.org/TR/xmlschema-1, http://www.w3.org/TR/xmlschema-2). This standard is being considered as a standard to be supported within NCES.
- MIME Encapsulation of Aggregate Documents such as HTML (MHTML), IETF RFC 2557 (to aggregate multi-resource documents in MIME-formatted messages). This standard is being considered as a standard to be supported within NCES.
- Web Distributed Authoring and Versioning (Web-DAV), IETF RFCs 2518, 3523. This standard is being considered as a standard to be supported within NCES.

8.3 XML Foundational

- XML Namespaces (Version 1.0: http://www.w3.org/TR/REC-xml-names, Version 1.1: http//www.w3.org/TR/xml-names11).

- XML Schema specification set:
  - XML Schema Part 0: Primer, http://www.w3.org/TR/xmlschema-0
  - XML Schema Part 1: Structures, http://www.w3.org/TR/xmlschemas-1
  - XML Schema Part 2: Datatypes, http://www.w3.org/TR/xmlschema-2
- Extensible Style Language Transformations (XSLT) (http://www.w3.org/TR/xslt). This is an emerging standard.
- Extensible Style Language (XSL) (http://www.w3.org/Style/XSL, http://www.w3.org/TR/xsl). This is an emerging standard.
- XML Path Language (XPath) (http://www.w3.org/TR/xpath). This is an emerging standard.
- Cascading Style Sheets (CSS) (http://www.w3.org/Style/CSS, http://www.w3.org/TR/REC-CSS1, http://www.w3.org/TR/REC-CSS2). This is an emerging standard.

8.4 Services Foundational

- Simple Object Access Protocol (SOAP) 1.1 (http://www.w3.org/TR/SOAP). This is an emerging standard.
- Web Services Description Language (WSDL) 1.1 (http://www.w3.org/2002/ws/desc, http://www.w3.org/TR/wsdl). This is an emerging standard.
- Universal Description, Discovery, and Integration (UDDI) 2.0 (http://www.uddi.org, http://www.oasis-open.org/committees/uddi-spec). This is an emerging standard.
- WS-Security (http://www.oasis-open.org/committees/wss).
- Web Services Interoperability (WS-I) Basic Profile specification (http://www.ws-i.org).

9 Mapping Guidance Actions to Enterprise Technology Objectives

This section maps the checklist guidance actions identified above to each of the enterprise technology objectives described in the *NESI Part 1: Net-Centric Overview*.

Section	Guidance Category	Capability On Demand	Distributed Operations	Customized Applications	Multi-User Access	Customized Delivery	Assured Sharing	Incremental Upgrade	Data Exchange
3	**Data**								
3.1	**Design tenet: Make data visible**								
3.1.1	General								X
3.1.2	DoD discovery metadata specification	X				X			X
3.1.3	Metadata generation	X				X			X
3.2	**Design tenet: Make data accessible**								
3.2.1	XML requirement	X				X			X
3.2.2	XML interface specification	X		X		X			X
3.2.3	XML interface usage	X		X		X			X
3.2.4	XML transport	X		X	X	X			X
3.2.5	Open-standard alternatives to XML format	X		X		X		X	X
3.2.6	Proprietary alternatives to XML format	X		X		X		X	X

Section	Guidance Category	Capability On Demand	Distributed Operations	Customized Applications	Multi-User Access	Customized Delivery	Assured Sharing	Incremental Upgrade	Data Exchange
3.3	**Design tenet: Make data understandable**								
3.3.1	XML schema usage	X		X		X			X
3.3.2	XML schema documentation	X		X		X		X	X
3.4	**Design tenet: Make data trustable**								
3.4.1	General	X				X	X		X
3.4.2	Authoritative source	X				X	X		X
3.4.3	Aggregated data	X				X	X		X
3.5	**Design tenet: Make data interoperable**								
3.5.1	XML wrapped data	X		X		X	X	X	X
3.5.2	XML schema validation	X				X	X		X
3.6	**Design tenet: Provide data management**								
3.6.1	General	X			X	X			X
3.7	**Design tenet: Be responsive to user needs**								
3.7.1	General	X			X	X			X
4	**Services**								
4.1	**Design tenet: Service-oriented architecture (SOA)**								
4.1.1	Service-oriented architecture	X	X	X	X			X	
4.1.2	Service description	X	X	X	X	X		X	

Section	Guidance Category	Capability On Demand	Distributed Operations	Customized Applications	Multi-User Access	Customized Delivery	Assured Sharing	Incremental Upgrade	Data Exchange
4.1.3	Service access point (SAP)	X	X	X	X			X	
4.1.4	Service design	X	X	X	X			X	
4.1.5	Service design characteristics	X	X	X	X			X	
4.1.6	Service implementation characteristics	X	X	X	X			X	
4.1.7	Service level agreement (SLA)	X	X	X	X	X		X	
4.1.8	Service interfaces	X	X	X	X			X	
4.1.9	Node responsibilities	X		X	X		X	X	
4.1.10	Service registration	X	X	X	X			X	
4.1.11	Service security	X	X	X	X		X	X	
4.1.12	Support for service orchestration	X	X	X	X	X		X	
4.2	**Design tenet: Open architecture**								
4.2.1	General	X	X	X	X			X	
4.2.2	Component based	X	X	X	X			X	
4.2.3	Public interfaces	X	X	X	X			X	
4.2.4	Layered software architecture	X	X	X	X			X	
4.2.5	Client tier	X	X	X	X			X	

Section	Guidance Category	Capability On Demand	Distributed Operations	Customized Applications	Multi-User Access	Customized Delivery	Assured Sharing	Incremental Upgrade	Data Exchange
4.2.6	Presentation tier	X	X	X	X			X	
4.2.7	Middle tier	X	X	X	X			X	
4.2.8	Data tier	X	X	X	X			X	X
4.2.9	Wrapping legacy systems	X	X	X	X			X	
4.3	**Design tenet: Scalability**								
4.3.1	General	X	X	X	X	X		X	
4.4	**Design tenet: Availability**								
4.4.1	General	X	X	X	X	X		X	
4.5	**Design tenet: Accommodate heterogeneity**								
4.5.1	Service structure	X	X	X	X			X	
4.5.2	Service configuration	X	X	X	X			X	
4.5.3	Node structure	X	X	X	X			X	
4.6	**Design tenet: Decentralized operations and management**								
4.6.1	General	X	X	X	X			X	
4.7	**Design tenet: Enterprise service management**								
4.7.1	Service management	X	X	X	X			X	
4.7.2	Provisioning of enterprise services	X	X	X	X			X	
5	**Information assurance/security**								
5.1	**Design tenet: Net-centric IA posture and continuity of operations**								

Section	Guidance Category	Capability On Demand	Distributed Operations	Customized Applications	Multi-User Access	Customized Delivery	Assured Sharing	Incremental Upgrade	Data Exchange
5.1.1	Mission assurance category	X			X		X		
5.2	**Design tenet: Identity management, authentication, and privileges**								
5.2.1	User authentication	X	X		X		X		
5.2.2	Identity management	X	X		X		X		
5.2.3	Multi-tier authentication	X	X		X		X		
5.2.4	Authorization processes	X			X		X		
5.2.5	Role-based authorizations	X	X		X		X	X	
5.2.6	Validation of authentication information	X	X		X		X	X	
5.3	**Design tenet: Mediate security assertions**								
5.3.1	Security assertions	X	X		X		X		
5.3.2	Chained requests	X	X		X		X		
5.4	**Design tenet: Cross-security-domains exchange**								
5.4.1	Cross-domain services	X	X		X		X		
5.5	**Design tenet: Encryption and HAIPE**								
5.5.1	Trusted paths establishment	X	X		X		X		
5.6	**Design tenet: Employment of wireless technologies**								

Section	Guidance Category	Capability On Demand	Distributed Operations	Customized Applications	Multi-User Access	Customized Delivery	Assured Sharing	Incremental Upgrade	Data Exchange
5.6.1	Wireless technologies	X	X		X	X	X		
5.7	**Other design tenets**								
5.7.1	Integrity and confidentiality	X			X		X		
5.7.2	Firewall configurations	X			X		X		
5.7.3	Intrusion detection systems	X			X		X		
5.7.4	Intrusion reporting	X			X		X		
5.7.5	Audit events linkage	X			X		X		
5.7.6	Use of audits for attribution	X			X		X		
5.7.7	GIG policy compliance	X			X		X		
5.7.8	Certification and accreditation	X			X		X		
6	**Transport**								
6.1	**Design tenet: IPv6**								
6.1.1	Support IPv6 transition	X	X		X	X		X	
6.1.2	Support IPv6 IP security features	X	X		X	X	X		
6.1.3	Implement DoD-adopted IPv6 standards and products	X	X		X	X			

Section	Guidance Category	Capability On Demand	Distributed Operations	Customized Applications	Multi-User Access	Customized Delivery	Assured Sharing	Incremental Upgrade	Data Exchange
6.2	**Design tenet: Packet switched infrastructure**								
6.2.1	Implement interface to one and only one network layer (layer-3) protocol for datagrams	X	X		X	X		X	
6.3	**Design tenet: Layering and modularity**								
6.3.1	Define layer boundaries and interfaces	X	X		X	X		X	
6.3.2	Ensure functions are modular and separable	X	X		X	X		X	
6.3.3	Minimize complexity of layered implementation	X	X		X	X		X	
6.4	**Design tenet: Transport goal**								
6.4.1	Support interfaces with converged traffic networks	X	X		X	X		X	
6.5	**Design tenet: Network connectivity**								
6.5.1	Manage scalability and complexity	X	X		X	X			
6.5.2	Optimize use of COTS products	X	X		X	X		X	
6.6	**Design tenet: Concurrent transport of information flows**								
6.6.1	Implement INE standards and	X	X		X	X	X		

Section	Guidance Category	Capability On Demand	Distributed Operations	Customized Applications	Multi-User Access	Customized Delivery	Assured Sharing	Incremental Upgrade	Data Exchange
	products to support traffic convergence								
6.6.2	Document approach to information infrastructure with black core	X	X		X	X	X		
6.7	**Design tenet: Differentiated management of quality of service**								
6.7.1	Support quality of service (QoS) and class of service (CoS)	X	X		X	X			
6.8	**Design tenet: Inter-network connectivity**								
6.8.1	Support internetwork connectivity using DOD-adopted standards	X	X		X	X		X	
6.9	**Design tenet: Joint technical architecture**								
6.9.1	Justify and document all standards that are not included in JTA version 6.0	X	X		X	X			
6.10	**Design tenet: RF acquisition**								
6.10.1	Justify, document, and obtain a waiver for all radio terminal acquisitions that are not JTRS/SCA compliant	X	X		X	X			

Section	Guidance Category	Capability On Demand	Distributed Operations	Customized Applications	Multi-User Access	Customized Delivery	Assured Sharing	Incremental Upgrade	Data Exchange
6.10.2	Minimize RF bandwidth capabilities	X	X		X	X			
6.11	**Design tenet: Joint net-centric capabilities**								
6.11.1	Employ NCOW reference model (NCOW RM)	X	X		X	X			
6.12	**Design tenet: Operations and management of transport and services**								
6.12.1	Develop manageable systems	X	X		X	X			
6.12.2	Use non-proprietary implementations	X	X		X	X			
6.12.3	Use accepted industry standards and emerging NetOps concepts	X	X		X	X			
6.12.4	Support standardized DoD service-oriented environment	X	X		X	X			
6.12.5	Employ DOD-adopted standards to support cross-system and domain management	X	X		X	X		X	
6.12.6	Plan for coalition interoperability	X	X	X	X	X			